these claims is not intended to be an acquiescence in the Office's assessment of those claims in the 25 May 1999 Communication, and applicants expressly reserve the right to bring the subject matter of the original claims again in a subsequent, related application.

Basis for the amendment to claim 11 can be found throughout the specification, for example, at the following locations: originally presented claim 11; and basis for "specifically binds" is discussed below.

Basis for the amendment to claim 38 can be found throughout the specification, for example, at the following locations: originally presented claim 38; and page 16, lines 22-29.

Basis for the amendment to claim 39 can be found throughout the specification, for example, at the following locations: originally presented claims 11 and 39; page 16, lines 22-29; and basis for "specifically binds" is discussed below.

Basis for new claims 45-48 can be found throughout the specification, for example, at the following location: page 13, line 34, to page 14, line 24.

Accordingly, no new matter has been added by way of this amendment and the entry thereof is respectfully requested.

Addressing the Examiner's Objections and Rejections

1. Objections to the Specification and Claims

The Examiner has objected to the following informalities in the present application: an incorrect address for the American Type Culture Collection; and, improper sequence number identifiers.

The applicants thank the Examiner for pointing out these informalities. The address of the American Type Culture Collection has been corrected. Further, the claims have been amended to include proper sequence number identifiers (i.e., SEQ ID NOS:). No new matter has been entered by way of these amendments.

2. Rejection of Claims 11-14, 33, 38 and 39 under 35 U.S.C. §112, Second Paragraph

The Examiner has rejected claims 11-14, 33, 38, and 39 under 35 U.S.C. §112, second paragraph, asserting that the claims are indefinite for failing to particularly point out and distinctly claim the subject matter which the applicant regards as the invention. The Examiner has asserted the following specific deficiencies in the claims.

A. Percent Identity

The Examiner asserts that recitation of "% identity" is vague and indefinite.

The applicants respectfully disagree with the Examiner's position. On page 12, line 26, to page 13, line 8, the applicants discuss the use of available programs for calculating identity or similarity between sequences, in particular the applicants state the following:

> "Two or more polynucleotide sequences can be compared by determining their "percent identity." Two or more amino acid sequences likewise can be compared by determining their "percent identity." The percent identity of two sequences, whether nucleic acid or peptide sequences, is the number of exact matches between two aligned sequences divided by the length of the shorter sequences and multiplied by 100. An approximate alignment for nucleic acid sequences is provided by the local homology algorithm of Smith and Waterman, Advances in Applied Mathematics 2:482-489 (1981). This algorithm can be extended to use with peptide sequences using the scoring matrix developed by Dayhoff, Atlas of Protein Sequences and Structure, M.O. Dayhoff ed., 5 suppl. 3:353-358, National Biomedical Research Foundation, Washington, D.C., USA, and normalized by Gribskov, Nucl. Acids Res. 14(6):6745-6763 (1986). An implementation of this algorithm for nucleic acid and peptide sequences is provided by the Genetics Computer Group (Madison, WI) in their BestFit utility application. The default parameters for this method are described in the Wisconsin Sequence Analysis Package Program Manual, Version 8 (1995) (available from Genetics Computer Group, Madison, WI)."

The applicants submit that use of default parameters is routine and well within the abilities of one having ordinary skill in the art. Further, accompanying this response the applicants are suppling a copy of the Wisconsin Sequence Analysis Package, Program Manual, Version 8, which describes the use of the "BestFit" utility application.

Absolute specificity and precision are not required in the claims. Claims need only reasonably apprize a person having ordinary skill in the art as to their scope. *Hybritech Inc., v. Monoclonal Antibodies, Inc.,* 231 USPQ 81, Fed. Cir. 1986. The second paragraph of 35 U.S.C. §112 merely requires that an applicant set out and circumscribe a particular subject area with a reasonable degree of precision such that the metes and bounds of the invention are set forth. *Ex parte Head,* 214 USPQ 551, PTO Bd. App. 1981.

However, in an effort to facilitate prosecution, applicants have deleted the language "percent identity" from the pending claims, as suggested by the Examiner. Applicants have introduced the language "specifically binds to," which replaces the language "specifically hybridizes to." The specification provides extensive basis for use of this language (i.e., specifically binds to). For example, on page 20, lines 18-34, detection of an analyte is discussed wherein a specific binding member is prepared for binding to a target analyte such as a nucleotide target. On page 21, lines 14-26, a definition of "specific binding members" is discussed, wherein a "specific binding member" is a member of a specific binding pair (see also, e.g., page 22, lines 1-16; page 23, line 32, to page 24, line 30; and page 5, line 1, to page 6, line 33). That is, two different molecules where one of the molecules, through chemical or physical means, specifically binds to the second molecule. Specific binding pairs can include complementary nucleotide sequences. On pages 27-34, the specification describes how the sequences provided in the application may be used to produce polynucleotide sequences (for example, primers and probes; also see, e.g., page 14, lines 25-32) which can be used in assays for the detection of target nucleic acids in test samples, via specifically binding the polynucleotide sequences to the target. Probes may, for example, be designed from conserved nucleotide regions of the polynucleotides of interest or from non-conserved nucleotide regions of the polynucleotide of interest. The design of such probes for optimization in assays is within the skill of the routineer. Generally, nucleic acid probes are developed from non-conserved or unique regions when maximum specificity is desired, and nucleic acid probes are developed from conserved regions when assaying for nucleotide regions that are closely related to, for example, different members of a multi-gene family or in related species like mouse and man. Numerous examples are given in the specification that would allow one of ordinary skill in the art to determine the metes and bounds of the invention (e.g., Examples 1-9, pages 57-69). For example, selection of primers for use in polymerase chain reactions is described at least on page 27, line 23, to page 28, line 32, and exemplary conditions (including hybridization conditions) for such reactions are described in the Examples (e.g., Examples 4, 8 and 9).

Use of probes in fluorescent in situ hybridization (FISH) technology to perform chromosomal analysis is also described herein. Such an approach can be used to identify cancer-specific structural alterations in the chromosomes, such as deletions or translocations that are visible from chromosome spreads or detectable using PCR-generated and/or allele specific oligonucleotides probes, allele specific amplification or by direct sequencing. Probes also can be labeled with radioisotopes, directly- or indirectly- detectable haptens, or fluorescent molecules, and utilized for *in situ* hybridization studies to evaluate the mRNA expression of the gene comprising the polynucleotide in tissue specimens or cells (page 27, lines 1-9; and Example 7, pages 66-67). Use of the polynucleotide sequences of the present invention in such technology is another example of specific binding of a polynucleotide sequence to a target.

The characteristics and properties of polynucleotides of the present invention for use in hybridization reactions (including probes and amplification primers) are extensively discussed in the specification in the context of specific binding (see, for example, pages 27-34). Further, examples using polynucleotides in hybridization reactions are discussed in the application, including suitable reaction conditions (e.g., Examples 5, 6, and 7, pages 64-67).

The court has consistently stated that claim language must be read in light of prior art and teachings of the specification. The standard is that the "definiteness of the language must be analyzed...in light of the teachings of the prior art and of the particular application disclosure as it would be interpreted by one possessing the ordinary level of skill in the pertinent art." *In re Moore*, 439 F.2d 1232, 169 USPQ 236 (CCPA 1971). A claim which is clear to one ordinarily skilled in the art when read in light of the specification, does not fail for indefiniteness. *Slimfold Mfg. Co. v. Kinkead Indus., Inc.*, 932 F2d 1453, 1 USPQ2d 1536 (Fed. Cir 1986).

In view of the above amendments, the teachings of the specification and the level of ordinary skill in the present art, the applicants submit that the boundaries of the claims are capable of being understood by one of ordinary skill in the art. Therefore, withdrawal of the rejection of the claims under 35 U.S.C. §112, second paragraph, is respectfully requested.

B. Fragment

The Examiner has objected to the term "fragment" asserting that it is not specifically defined in the specification or the claims. The applicants respectfully traverse this assertion. The term "fragment" is defined in the specification at least at page 14, lines 19-24. However, in order to facilitate prosecution the applicants have removed the term "fragment" from the claims and have instead recited specific sizes of polynucleotides, where the polynucleotides are limited by their ability to specifically bind the reference sequence. Accordingly, withdrawal of the rejection of the claims under 35 U.S.C. §112, second paragraph, is respectfully requested.

C. Capable of Specifically Hybridizing

The Examiner has objected to the phrase "capable of specifically hybridizing" and suggests that it be replaced by "specifically hybridizes to." The applicants have removed the language "capable of" and have replaced "specifically hybridizes" with --specifically binds--. This amendment was further discussed above. Accordingly, withdrawal of the rejection of the claims under 35 U.S.C. §112, second paragraph, is respectfully requested.

D. Claim 33

Claim 33 has been canceled, without prejudice or disclaimer, in order to facilitate prosecution.

In view of the above amendments and comments the applicants submit that the claims comply with the requirements of 35 U.S.C. §112, second paragraph. Accordingly, withdrawal of the rejection of the claims is respectfully requested.

3. Rejection of Claims 11-14 and 33 Under 35 U.S.C. §102(b)

The Examiner has rejected claims 11-14 and 33 under 35 U.S.C. §102(b) asserting that the claims are clearly anticipated by Hillier, et al.

The Examiner asserts that SEQ ID NOS:4, 5, 7 and 12 are anticipated. In order to facilitate prosecution applicants have deleted these sequences from the claims. This amendment of the claims is not intended to be an acquiescence in the Office's assessment of those claims. In view of the above amendments and arguments, the cited reference sequences cannot be said to teach all the elements of the present invention. The dependent claims distinguish over the prior art at least in view of their dependencies on the independent claims. Accordingly, the applicants submit that the pending claims are not anticipated by the cited prior art under 35 U.S.C. §102(b) and withdrawal of the rejection is respectfully requested.

CONCLUSION

Applicant respectfully submits that the claims comply with the requirements of 35 U.S.C. §112 and define an invention that is patentable over the art. Accordingly, a Notice of Allowance is believed in order and is respectfully requested.

If the Examiner notes any further matters which the Examiner believes may be expedited by a telephone interview, the Examiner is requested to contact the undersigned.

Please direct all further communications in this application to:

Cheryl L. Becker, Esq.
Abbott Laboratories
D-377/AP6D-2
100 Abbott Park Road
Abbott Park, IL 60064-3500
Telephone: (847) 935-1729
Facsimile: (847) 938-2623

Respectfully submitted,

Date: 27 Sept 1999 By: Gary R Fabian
Gary R. Fabian, Ph.D.
Registration No. 33,875
Agent for Applicants

enclosure: pages of the Wisconsin Sequence Analysis Package Program Manual

ABBOTT LABORATORIES
D-377/AP6D-2, 100 Abbott Park Road
Abbott Park, IL 60064-3500
Telephone: (847) 935-1729
Facsimile: (847) 938-2623

Wisconsin Sequence Analysis Package

Program Manual

▶ Version 8

UNIX

Genetics Computer Group
University Research Park
575 Science Drive
Madison, WI 53711

Telephone Number: (608)231–5200
FAX: (608)231–5202
E–Mail Technical Support: Help@GCG.Com

GCG

Copyright

Licenses and Trademarks

Credits

Technical editors: Irv Edelman, Sue Olson, and John Devereux.

Cover and tab graphic design: Andrew Rouse, Maggie Smith, Dina Beers, and Joleen Rau.

Production: Ann Kiefer and Dawn Stencil.

Acknowledgements

Excerpt from *Now, Where Were We?* Copyright 1989 by Ray Blount by permission of Ray Blount. Excerpt from *Nature* (Volume 171:737–738, 1953). Copyright 1953 by Macmillan Magazines Limited, by permission of Macmillan Magazines Limited. Excerpt from *Annals of the New York Academy of Sciences* (Volume 69). Copyright 1957 by New Y rk Academy of Sciences, by permission of New York Academy of Sciences. Quote by Carl R. Woese, by permission of Carl R. Woese. Excerpt from *The Gene Wars: Science, Politics, and the Human Genome* by Robert Cook–Deegan. Copyright 1994 by Robert Cook–Deegan, by permission of Robert Cook–Deegan. Quotes by Edward O. Wilson, by permission of Edward O. Wilson. Excerpt from *Margot Fonteyn:Autobiography.* Copyright 1975 by Knopf Publishers, by permission of Knopf Publishers. Excerpts from *A Passion for Science*, edited by Lewis Wolpert and Alison Richards. Copyright by Oxford University Press, by permission of Oxford University Press. Excerpt from *The Wisdom of the Genes.* Copyright 1989 by Christopher Wills, by permission of Christopher Wills. Excerpt from *Molecules and Evolution.* Copyright 1966 by Columbia University Press, by permission of Columbia University Press.

Document Revision History

Software Version	*Revision Date*	*Principal Changes*
8.0	September 1994	Added new programs: BLAST, ToBLAST, Prime, Distances, GrowTree, NoOverlap, and FromFastA. See the What's New in Version 8.0 release notes for more information.
8.1	August 1995	Added new programs: FrameAlign, FrameSearch, LookUp, and NewDiverge. Updated BLAST. See the What's New in Version 8.1 release notes for more information.

BESTFIT

FUNCTION

BestFit makes an optimal alignment of the best segment of similarity between two sequences. Optimal alignments are found by inserting gaps to maximize the number of matches using the *local homology* algorithm of Smith and Waterman.

DESCRIPTION

BestFit inserts gaps to obtain the optimal alignment of the best region of similarity between two sequences, and then displays the alignment in a format similar to the output from Gap. The sequences can be of very different lengths and have only a small segment of similarity between them. You could take a short RNA sequence, for example, and run it against a whole mitochondrial genome.

SEARCHING FOR SIMILARITY

BestFit is the most powerful method in the Wisconsin Sequence Analysis Package™ for identifying the best region of similarity between two sequences whose relationship is unknown.

EXAMPLE

The sequence gamma.seq contains an Alu family sequence somewhere in the first 500 bases. alu.seq contains a generic human Alu family repeat. The two sequences are aligned and the best segment of similarity is found with BestFit.

```
% bestfit

 BESTFIT of what sequence 1 ?  gamma.seq

                  Begin (* 1 *) ?
                End (* 11375 *) ?  500
              Reverse (* No *) ?

 to what sequence 2 (* gamma.seq *) ?  alu.seq

                  Begin (* 1 *) ?
                End (*   207 *) ?
              Reverse (* No *) ?

 What is the gap creation penalty (* 5.00 *) ?

 What is the gap extension penalty (* 0.30 *) ?

 What should I call the paired output display file (* gamma.pair *)

 Aligning ..........-..

          Gaps:     3
       Quality: 129.3
 Quality Ratio: 0.625
  % Similarity: 84.466
        Length:   209

%
```

OUTPUT

Here is the output file. Notice how BestFit finds and displays only the best segments of similarity:

```
 BESTFIT of: gamma.seq  check: 6474  from: 1  to: 500

Human fetal beta globins G and A gamma
from Shen, Slightom and Smithies,  Cell 26; 191-203.
Analyzed by Smithies et al. Cell 26; 345-353.

 to: alu.seq  check: 4238  from: 1  to: 207

HSREP2 from the EMBL data library
Human Alu repetitive sequence located near the insulin gene
Dhruva D.R., Shenk T., Subramanian K.N.; "Integration in vivo into
Simian virus 40 DNA of a sequence that resembles a certain family of
genomic interspersed repeated sequences"; Proc. Natl. Acad. Sci. USA
77:4514-4518(1980). . . .

 Symbol comparison table: Gencoredisk:[Gcgcore.Data.Rundata]Swgapdna.Cmp
 CompCheck: 5234

         Gap Weight:  5.000      Average Match:  1.000
      Length Weight:  0.300   Average Mismatch: -0.900

            Quality:  129.3             Length:    209
              Ratio:  0.625               Gaps:      3
 Percent Similarity: 84.466   Percent Identity: 84.466

 gamma.seq x alu.seq       June 20, 1994 15:15  ..


                 .         .         .         .         .
    137 AGACCAACCTGGCCAACATGGTGAAATCCCATCTCTAC.AAAAATACAAA 185
        |||||| |||||||||||||||||||  ||||||||||  ||||||||||  45
      1 AGACCAGCCTGGCCAACATGGTGAAACTCCATCTCTACTGAAAATACAAA 50

                 .         .         .         .         .
    186 AATTAGACAGGCATGATGGCAAGTGCCTGTAATCCCAGCTACTTGGGAGG 235
        |||||| |||||||| ||    ||||||| |||||||||||||| |||||  41
     51 AATTAGCCAGGCATGGTGATGCGTGCCTGGAATCCCAGCTACTTAGGAGG 100

                 .         .         .         .         .
    236 CTGAGGAAGGAGAATTGCTTGAACCTGGAAGGCAGGAGTTGCAGTGAGCC 285
        |||||  || |||||  ||| ||||  | |||  |  |||||||||||||
    101 CTGAGACAGAAGAATCCCTTAAACCAAG.AGGTGGAGGTTGCAGTGAGCC 149  37

                 .         .         .         .         .
    286 GAGATCATACCACTGCACTCCAGCCTGGGTGACAGAACAAGACTCTGTCT 335
        ||||||  ||  |||||||||||||| ||||||||| | ||||||  |||  41
    150 GAGATCGCACGGCTGCACTCCAGCCT.GGTGACAGAGCGAGACTCCATCT 198

    336 CAAAAAAAA 344
        |||||||||
    199 CAAAAAAAA 207    9
```

RELATED PROGRAMS

When you want an alignment that covers the whole length of both sequences, use Gap. When you are trying to find only the best segment of similarity between two sequences, use BestFit. PileUp creates a multiple sequence alignment of a group of related sequences, aligning the whole length of all sequences. DotPlot displays the entire surface of comparison for a comparison of two sequences. GapShow displays the pattern of differences between two aligned sequences. PlotSimilarity plots the average similarity of two or more aligned sequences at each position in the alignment. Pretty displays alignments of several sequences. LineUp is an editor for editing multiple sequence alignments. CompTable helps generate scoring matrices for peptide comparison.

ALGORITHM

BestFit uses the *local homology* algorithm of Smith and Waterman (Advances in Applied Mathematics 2; 482-489 (1981)) to find the best segment of similarity between two sequences. BestFit reads a scoring matrix that contains values for every possible GCG symbol match (see the LOCAL DATA FILES topic below). The program uses these values to construct a path matrix that represents the entire surface of comparison with a score at every position for the best possible alignment to that point. The *quality* score for the best alignment to any point is equal to the sum of the scoring matrix values of the matches in that alignment, less the gap creation penalty times the number of gaps in that alignment, less the gap extension penalty times the total length of all gaps in that alignment. The gap creation and gap extension penalties are set by you. If the best path to any point has a negative value, a zero is put in that position.

After the path matrix is complete, the highest value on the surface of comparison represents the end of the best region of similarity between the sequences. The best path from this highest value backwards to the point where the values revert to zero is the alignment shown by BestFit. This alignment is the best segment of similarity between the two sequences.

For nucleic acids, the default scoring matrix has a *match* value of 1.0 for each identical symbol comparison and -0.90 for each non-identical comparison (not considering nucleotide ambiguity symbols for this example). The *quality* score for a nucleic acid alignment can, therefore, be determined using the following equation:

$$\begin{aligned}\mathtt{Quality} = {} & \mathtt{1.0 \times TotalMatches} + \mathtt{-0.90 \times TotalMismatches}\\ & - \mathtt{(GapCreationPenalty \times GapNumber)}\\ & - \mathtt{(GapExtensionPenalty \times TotalLengthOfGaps)}\end{aligned}$$

The *quality* score for a protein alignment is calculated in a similar manner. However, while the default nucleic acid scoring matrix has a single value for all non-identical comparisons, the default protein scoring matrix has different values for the various non-identical amino acid comparisons. The *quality* score for a protein alignment can therefore be determined using the following equation (where $\mathtt{Total}_{AA}$ is the total number of A-A (Ala-Ala) matches in the alignment, $\mathtt{CmpVal}_{AA}$ is the value for an A-A comparison in the scoring matrix, $\mathtt{Total}_{AB}$ is the total number of A-B (Ala-Asx) matches in the alignment, $\mathtt{CmpVal}_{AB}$ is the value for an A-B comparison in the scoring matrix, ...) :

$$\begin{aligned}\mathtt{Quality} = {} & \mathtt{CmpVal}_{AA} \times \mathtt{Total}_{AA}\\ & + \mathtt{CmpVal}_{AB} \times \mathtt{Total}_{AB}\\ & + \mathtt{CmpVal}_{AC} \times \mathtt{Total}_{AC}\\ & \quad\vdots\\ & + \mathtt{CmpVal}_{ZZ} \times \mathtt{Total}_{ZZ}\\ & - \mathtt{(GapCreationPenalty \times GapNumber)}\\ & - \mathtt{(GapExtensionPenalty \times TotalLengthOfGaps)}\end{aligned}$$

For a more complete discussion of scoring matrices, see the **Data Files** manual.

CONSIDERATIONS

BestFit Always Finds Something

BestFit always finds an alignment for any two sequences you compare – even if there is no significant similarity between them! You must evaluate the results critically to decide if the segment shown is not just a random region of relative similarity.

Th Segments Shown Obscure Alternative Segments

BestFit only shows one segment of similarity, so if there are several, all but one is obscured. You can approach this problem with graphic matrix analysis (see the Compare and DotPlot programs). Alternatively, you can run BestFit on ranges outside the ranges of similarity found in earlier runs to bring other segments out of the shadow of the best segment.

The Best Fit is Only One Member of a Family

Like all fast gapping algorithms, the alignment displayed is a member of the family of best alignments. This family may have other members of equal quality, but will not have any member with a higher quality. The family is usually significantly different for different choices of gap creation and gap extension penalties. See the CONSIDERATIONS topic in the entry for the Gap program in the **Program Manual** to learn more about how to assign gap creation and gap extension penalties.

The Surface of Comparison

The magnitude of the computer's job is proportional to the area of the surface of comparison. That area is determined by the product of the lengths of the two sequences compared. BestFit can evaluate a surface of up to 3.5 million elements. This surface would be large enough to compare two sequences approximately 1,870-symbols long, or one sequence 200-symbols long with another sequence 17,500-symbols long. When you have much longer sequences that are known to align well, you can use the command-line option `-LIMit` to use the surface more efficiently.

Th Public Scoring Matrix for Nucleic Acid Comparisons is Very Stringent

The scoring matrix swgapdna.cmp penalizes mismatches -0.9 so the segments found may be very brief. This penalty means that the alignment cannot be extended by three bases to pick one extra match. The scoring matrix used by Smith and Waterman, when local alignments were first described, used -0.333 for the mismatch penalty. You can use Fetch to copy randomdna.cmp and rename it swgapdna.cmp to use these values, or use nwsgapdna.cmp, which has no mismatch penalty at all.

Rapid Alignment

When possible, BestFit tries to find the optimal alignment very quickly. If this rapid alignment is not unambiguously optimal, BestFit automatically realigns the sequences to calculate the optimal alignment. When this occurs, the monitor of alignment progress on your terminal screen (`Aligning...`) is displayed twice for a single alignment.

ALIGNING LONG SEQUENCES

This program can align very long sequences if you know roughly where the alignment of interest begins. Run the program with the command line option `-LIMit`. Then set the starting coordinates for each sequence near the point where the alignment of interest begins and set gap shift limits on each sequence. The program then aligns the sequences from your starting point such that the sequences do not get out of phase by more than the gap shift limits you have set. If you started both sequences at base number one and set the gap shift limit for sequence one to 100 and for sequence two to 50, then base 350 in sequence one could not be gapped to any base outside of the range from 300 to 450 on sequence two.

If you omit `-LIMit` on the command line, the program automatically sets gap shift limits if they are needed to allow the alignment of long sequences to proceed. In this case, the program limits the total length of gaps that can be inserted into each sequence and calculates the best alignment within this incomplete, or *limited*, surface of comparison. The program then performs a calculation to determine whether the alignment could possibly be improved if there were no restriction on the total length of gaps in each sequence. If the program cannot rule out this possibility, it displays the message `*** Alignment is not guaranteed to be optimal ***`. Because the criteria used in the calculation for guaranteeing an optimal alignment are very stringent, a limited alignment often may be optimal even if this message is displayed. In any event, the program continues to completion.

EVALUATING ALIGNMENT SIGNIFICANCE

This program can help you evaluate the significance of the alignment, using a simple statistical method, with the `-RANdomizations` command line option. The second sequence is repeatedly shuffled, maintaining its length and composition, and then realigned to the first sequence. The average alignment score, plus or minus the standard deviation, of all randomized alignments is reported in the output file. You can compare this average *quality* score to the quality score of the actual alignment to help evaluate the significance of the alignment. The number of randomizations can be specified along with the `-RANdomizations` command line qualifier; the default is 10.

The score of each randomized alignment is reported to the screen. You can use `<Ctrl>C` to interrupt the randomizations and output the results from those randomized alignments that have been completed.

By ignoring the statistical properties of biological sequences, this simple Monte Carlo statistical method may give misleading results. Please see Lipman, D.J, Wilbur, W.J., Smith, T.F., and Waterman, M.S. (Nucl. Acids Res. 12; 215-226 (1984)) for a discussion of the statistical significance of nucleic acid similarities.

ALIGNMENT METRICS

BestFit and Gap display four figures of merit for alignments: Quality, Ratio, Identity, and Similarity.

The Quality (described above) is the metric maximized in order to align the sequences. Ratio is the quality divided by the number of bases in the shorter segment. Percent Identity is the percent of the symbols that actually match. Percent Similarity is the percent of the symbols that are similar. Symbols that are across from gaps are ignored. A similarity is scored when the scoring matrix value for a pair of symbols is greater than or equal to 0.50, the *similarity threshold*. This threshold is also used by the display procedure to decide when to put a ':' (colon) between two aligned symbols. You can reset it from the command line with the second optional parameter of `-PAIr`. For instance, the expression `-PAIr=1.0,0.5` would set the similarity threshold to 0.5.

The similarity and identity metrics are not optimized by alignment programs so they should not be used to compare alignments.

PEPTIDE SEQUENCES

If your input sequences are peptide sequences, this program uses a scoring matrix with matches scored as 1.5 and mismatches scored according to the evolutionary distance between the amino acids as measured by Dayhoff and normalized by Gribskov (Gribskov and Burgess Nucl. Acids Res. 14(16); 6745-6763 (1986)).

RESTRICTIONS

Input sequences may not be more than 30,000-symbols long. This program cannot evaluate a surface of comparison larger than 5.5 million elements. A 200 x 27,500 comparison is possible, as well as a 2,300 x 2,300 comparison. See the ALIGNING LONG SEQUENCES topic for help in aligning long sequences that would normally exceed the maximum surface of comparison. You can also ask your system manager to increase the maximum surface of comparison if your system has enough virtual memory.

SEQUENCE TYPE

The function of BestFit depends on whether your input sequence(s) are protein or nucleotide. Normally the type of a sequence is determined by the presence of either `Type: N` or `Type: P` on the last line of the text heading just above the sequence itself. If your sequence(s) are not the correct type, turn to Appendix VI for information on how to change or set the type of a sequence.

COMMAND-LINE SUMMARY

All parameters for this program may be put on the command line. Use the option `-CHEck` to see the summary below and to have a chance to add things to the command line before the program executes. In the summary below, the capitalized letters in the qualifier names are the letters that you *must* type in order to use the parameter. Square brackets ([and]) enclose qualifiers or parameter values that are optional. For more information, see "Using Program Parameters" in Chapter 3, Basic Concepts: Using Programs in the **User's Guide**.

```
Minimal Syntax: % bestfit [-INfile1=]gamma.seq  [-INfile2=]alu.seq -Default

Prompted Parameters:

-BEGin1=1 -BEGin2=1      beginning of each sequence
-END1=500 -END2=207      end of each sequence
-NOREV1     -NOREV2      strand of each sequence
-GAPweight=5.0           gap creation penalty    (3.0 is protein default)
-LENgthweight=0.3        gap extension penalty   (0.1 is protein default)
[-OUTfile1=]gamma.pair   output file for alignment

Local Data Files: -DATa=swgapdna.cmp  scoring matrix for nucleic acids
                  -DATa=swgappep.cmp  scoring matrix for peptides

Optional Parameters:

-OUTfile2=gamma.gap     new sequence file for sequence 1 with gaps added
-OUTfile3=alu.gap        "     "      "    "     "    2   "    "     "
-LIMit1=499 -LIMit2=206 limit the surface of comparison.
-RANdomizations[=10]    determine average score from 10 randomized
                            alignments
-PAIr=1.0,0.5,0.1       thresholds for displaying '|', ':', and '.'
-WIDth=50               the number of sequence symbols per line
-PAGe=60                adds a line with a form feed every 60 lines
-NOBIGGaps              suppresses abbreviation of large gaps with '.'s
-HIGhroad               makes the top alignment for your parameters
-LOWroad                makes the bottom alignment for your parameters
-NOSUMmary              suppresses the screen summary
```

ACKNOWLEDGEMENTS

Gap and BestFit were originally written for Version 1.0 by Paul Haeberli from a careful reading of the Needleman and Wunsch (J. Mol. Biol. 48; 443-453 (1970)) and the Smith and Waterman (Adv. Appl. Math. 2; 482-489 (1981)) papers.

Limited alignments were designed by Paul Haeberli and added to the Package for Version 3.0. They were united into a single program by Philip Delaquess for Version 4.0. Default gap penalties for protein alignments were modified according to the suggestions of Rechid, Vingron and Argos (CABIOS 5; 107-113 (1989)).

LOCAL DATA FILES

The files described below supply auxiliary data to this program. The program automatically reads them from a public data directory unless you either 1) have a data file with exactly the same name in your current working directory; or 2) name a file on the command line with an expression like `-DATa1=myfile.dat`. For more information see Chapter 4, Using Data Files in the **User's Guide**.

If the first sequence you name is a nucleic acid, BestFit uses the scoring matrix in the public file swgapdna.cmp. (SW stands for Smith and Waterman.) If the first sequence you name is a peptide sequence, BestFit reads swgappep.cmp instead. The presence of these files in your current working directory causes BestFit to read your version instead. (See the **Data Files** manual for more information about scoring matrices.)

OPTIONAL PARAMETERS

The parameters and switches listed below can be set from the command line. For more information, see "Using Program Parameters" in Chapter 3, Basic Concepts: Using Programs in the **User's Guide**.

```
-LIMit1=20 and -LIMit2=20
```

let you set *gap shift limits* for each sequence. When you already know of a long similarity between two sequences you can "zip" them together using this mode. The beginning coordinates for each sequence must be near the beginning of the alignment you want to see. The alignment continues so that gaps inserted do not require the sequences to get out of step by more than the gap shift limits. You can align very long sequences rapidly. The surface of comparison is still limited to 3.5 million. The size of a comparison can be predicted by multiplying the average length of the two sequences by the sum of the two shift limits.

If you add `-LIMit` to the command line without any qualifier value, the program prompts you to enter gap shift limits for each sequence.

```
-RANdomizations=10
```

reports the average alignment score and standard deviation from 10 randomized alignments in which the second sequence is repeatedly shuffled, maintaining the length and composition of the original sequence, and then aligned to the first sequence. You can use the optional parameter to set the number of randomized alignment to some number other than 10.

```
-OUTfile2=seqname1.gap -OUTfile3=seqname2.gap
```

This program can write three different output files. The first displays the alignment of sequence one with sequence two. The second is a new sequence file for sequence one, possibly expanded by gaps to make it align with sequence two. The third, like the second, is a new sequence file for sequence two, possibly expanded by gaps to make it align with sequence one. The program writes only the first file unless there are output file options n the command line. If there are any output files named on the command line, *only* those output files are written. If you add

-OUT to the command line without any qualifying filename, then the program will write the second and third output files after prompting you for their names.

Aligned sequences (in sequence files) can be displayed with GapShow. Their similarity can be displayed with PlotSimilarity.

-PAIr=1.0,0.5,0.1

The paired output file from this program displays sequence similarity by printing one of three characters between similar sequence symbols: a pipe character(|), a colon (:), or a period (.). Normally a pipe character is put between symbols that are the same, a colon is put between symbols whose comparison value is greater than or equal to 0.50, and a period is put between symbols whose comparison value is greater than or equal to 0.10. You can change these *match display thresholds* from the command line. The three parameters for -PAIr are the display thresholds for the pipe character, colon, and period. The match display criterion for a pipe character changes from symbolic identity (the default) to the quantitative threshold you have set in the first parameter. A pipe character will no longer be inserted between identical symbols unless their comparison values are greater than or equal to this threshold. If you still want a pipe character to connect identical symbols, use x instead of a number as the first parameter. (See the **Data Files** manual for more information about scoring matrices.)

-PAGe=64

When you print the output from this program, it may cross from one page to another in a frustrating way – especially when you print on individual sheets. This option adds form feeds to the output file in order to try to keep clusters of related information together. You can set the number of lines per page by supplying a number after the -PAGe qualifier.

-WIDth=50

puts 50 sequence symbols on each line of the output file. You can set the width to anything from 10 to 150 symbols.

-NOBIGGaps

suppresses large gap abbreviations, showing all the sequence characters across from large gaps. Usually, gaps that extend one sequence by more than one complete line of output are abbreviated with three dots arranged in a vertical line.

-LOWroad and -HIGhroad

The insertion of gaps is, in many cases, arbitrary, and equally optimal alignments can be generated by inserting gaps differently. When equally optimal alignments are possible, this program can insert the gaps differently if you select either the -LOWroad or the -HIGhroad options. Here are examples for the alignment of GACCAT with GACAT with different parameters.

```
For:         Match = 1.0          MisMatch = -0.9
        Gap weight = 1.0    Length Weight =  0.0

LowRoad:     1 GACCAT 6
               || |||       Quality = 4.0
             1 GA.CAT 5

HighRoad:    1 GACCAT 6
               ||| ||       Quality = 4.0
             1 GAC.AT 5
```

```
For:        Match = 1.0        MisMatch =  0.0
      Gap weight = 3.0    Length Weight =  0.0

HighRoad:   1 GACCAT 6
               |||          Quality = 3.0
            1 GACAT. 5

LowRoad:    1 GACCAT 6
                  |||       Quality = 3.0
            1 .GACAT 5
```

Essentially the *low road* shifts all of the arbitrary gaps in sequence two to the left and all of the arbitrary gaps in sequence one to the right. The *high road* does exactly the opposite. When neither *high road* nor *low road* is selected, the program tries not to insert a gap whenever that is possible and uses the high road alternative for all collisions.

`-SUMmary`

writes a summary of the program's work to the screen when you've used the `-Default` qualifier to suppress all program interaction. A summary typically displays at the end of a program run interactively. You can suppress the summary for a program run interactively with `-NOSUMmary`.

Use this qualifier also to include a summary of the program's work in the log file for a program run in batch.